Objective Exam

Software Requirements Specification

Submitted by
Shreya Patel (B16CS025)
Saksham Gupta (B16CS030)

Revision History

Date	Version	Description	People
17 Feb 2018	1.0	Created Use-case and Sequence diagrams	Shreya Patel
		Created Class and Activity diagrams	Saksham Gupta
5 March 2018	2.0	Corrected Use-case diagram and changes in formatting	Shreya Patel Saksham Gupta

Table of Contents

1. Introduction 4

1.1 Purpose 4
1.2 Scope 4
1.3 Constraints 4
1.4 Assumptions and Dependencies 4
1.5 Definitions, Acronyms and Abbreviations 5
1.6 References 5
1.7 Organization of the Document 5

2. Overall Description 6

2.1 Product Functions 6
2.2 User Characteristics 12

3. Specific Requirements 12

3.1 Use case description 12
3.2 Supportability 15

Software Requirements Specification

1. *Introduction*

1.1 Purpose

The purpose of this document is to illustrate the requirements and describe the development and operations of the software- Objective Exam. This project aims at creating an objective exam portal for students to attempt an exam (quiz) and view the result as well as adding or editing the questions in the database by the faculty members. This would be useful for educational institutes for evaluating students by reducing time consumed and paperwork for manual examination.

1.2 Scope

The software aims at providing an objective exam portal for the students to attempt the exam as well as a convenient way for the faculties to generate and manage database of questions.

Scope of Faculty

- The faculty will be able to login using Login Id and Password.
- The faculty will be able to add questions, their options and correct answers alongwith the positive and negative marks associated with that particular question in the database.
- Only the faculty will have the access to the database of questions.

Scope of Student

- He will be able to register and login using login Id and Password.
- He will be able to take the test of different courses and get Score Sheet.

This software will thus help to reduce the hassle involved in taking exam on paper and the delay in displaying the results. Moreover the database for the questions can be managed easily for different courses.

1.3 Constraints

1. The number of questions in a test will be limited and the test will be time-bounded.
2. Student can attempt only one course test at a time. He cannot view others' profile or results.

1.4 Assumptions and Dependencies

1. Users should provide legitimate details while logging in or signing up.
2. Students will attempt the tests of his registered courses only.
3. The faculties are fixed, that is, no new faculty can be registered.
4. The database of questions should contain atleast 10 questions of each course.
5. The server where database is stored should be well secured.
6. The end user has proper understanding about functionality of the software.
7. All the hardware and software requirements are provided to the system so as to run this software.

1.5 Definitions, Acronyms and Abbreviations

Term	Definition
User	Includes both student and faculty
Student	User mostly a student who will attempt the test
Faculty	User mostly professors who will manage the database of questions

1.6 References

1) Sommerville, Software Engineering
2) Roger S. Pressman, Software Engineering: A Practitioner's Approach
3) UML@Classroom: An Introduction to Object-Oriented Modeling
4) Wikipedia – en.wikipedia.org/wiki

1.7 Organization of the Document

1. Section 2.1.1 talks about the functional requirements or the capabilities of the software and the functions that different users can perform through the software.
2. Section 2.1.2 describes the other system hardware requirements or other requirements that must be met for the software to run efficiently and effectively.
3. Section 2.2 gives brief description of the different users interacting with the software.

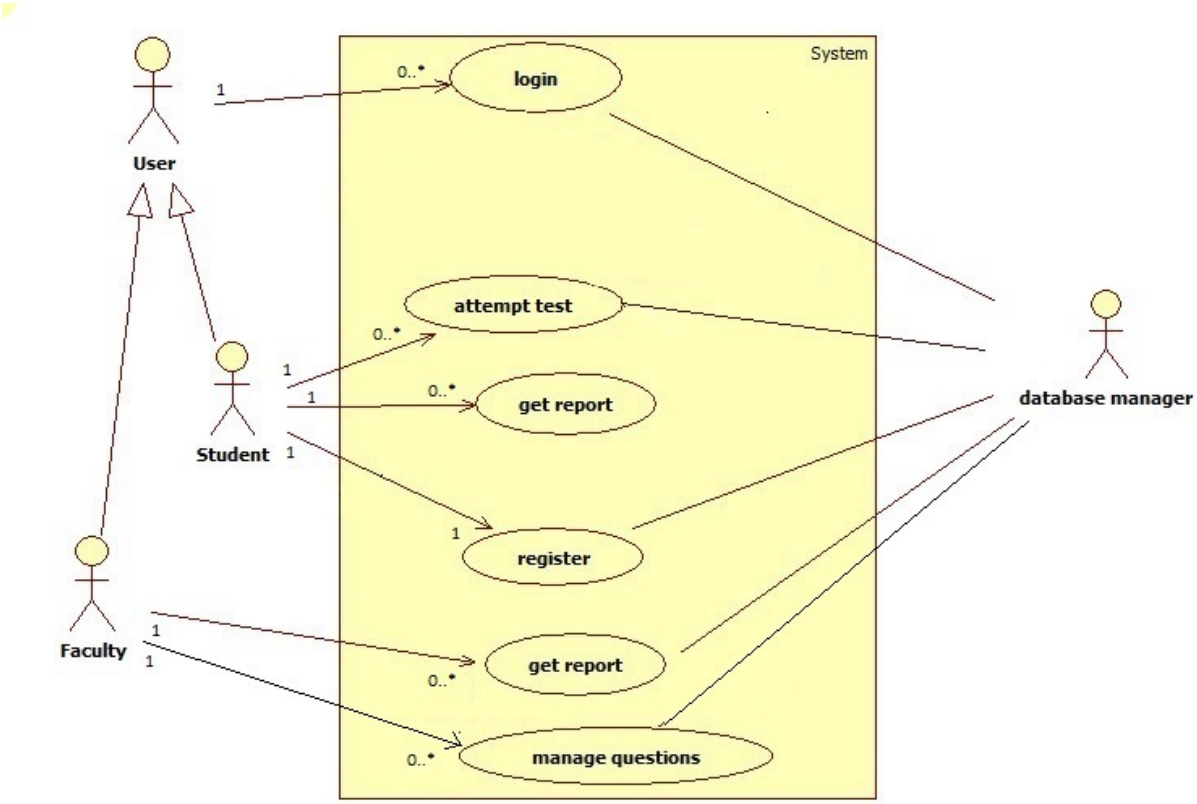
2. *Overall Description*

2.1 Product Functions

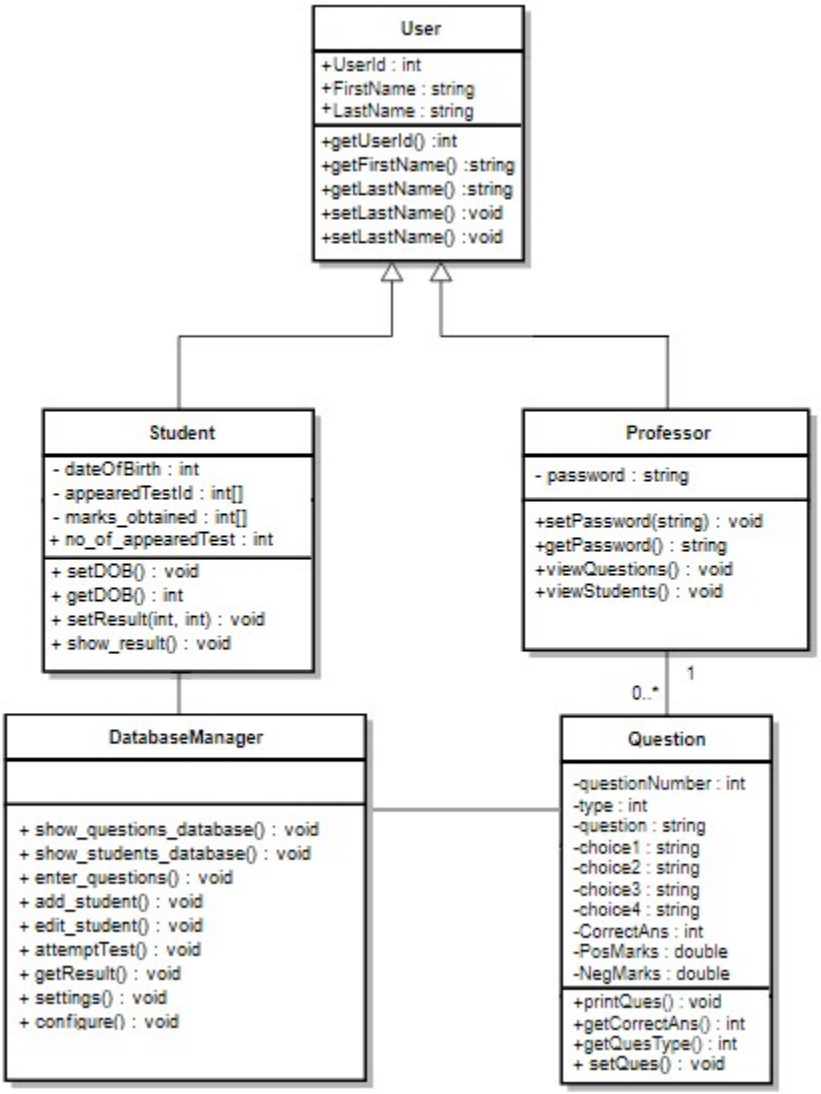
2.1.1 Functional Requirements

- All the users, both students and faculty, can login into the system.
- The faculty can add, edit or delete the questions of his course from the database.
- Students can select a course from the list of courses and can attempt the MCQs test.
- The student can view the Score sheet after conducting the test.

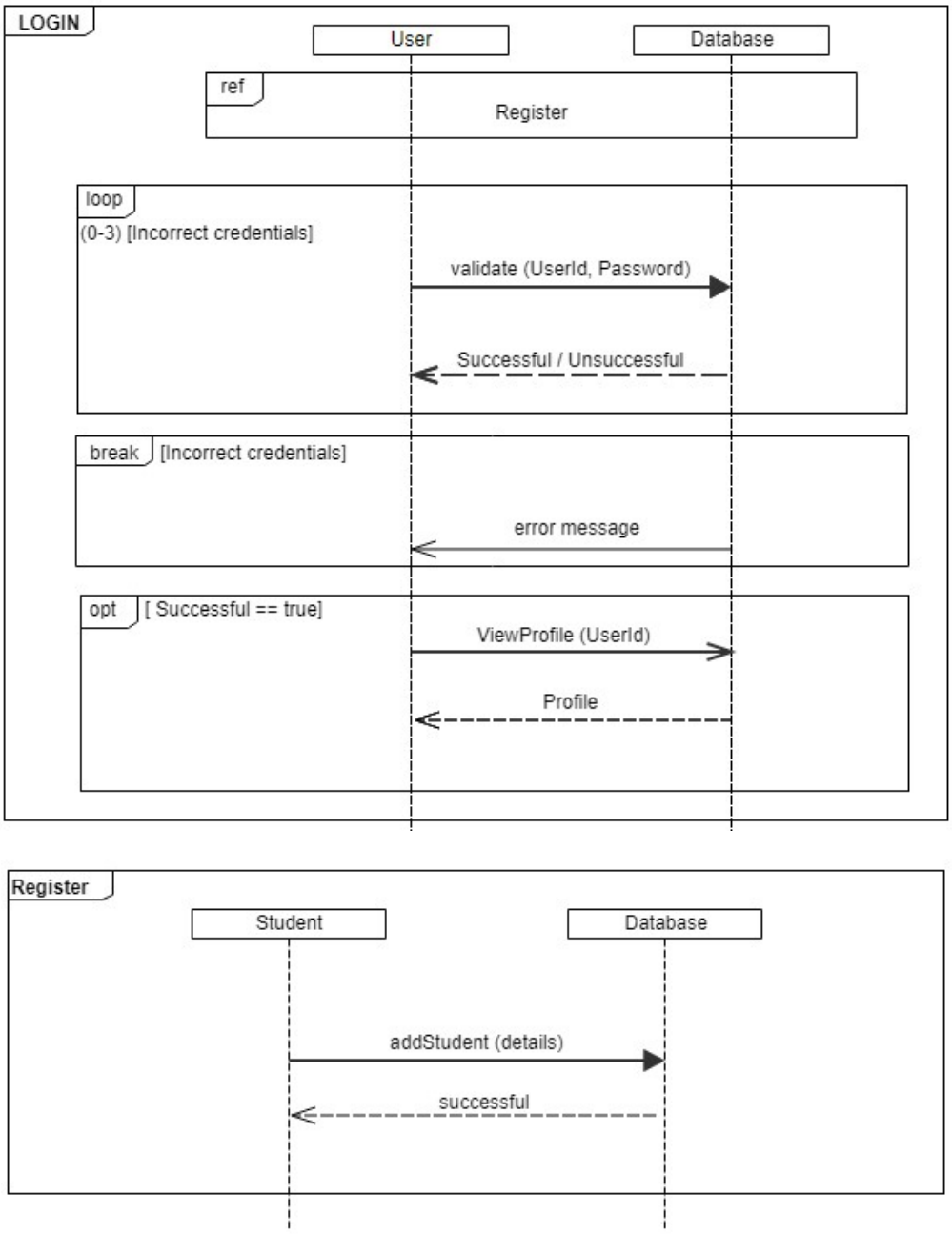
2.1.1.1 Use Case Diagram

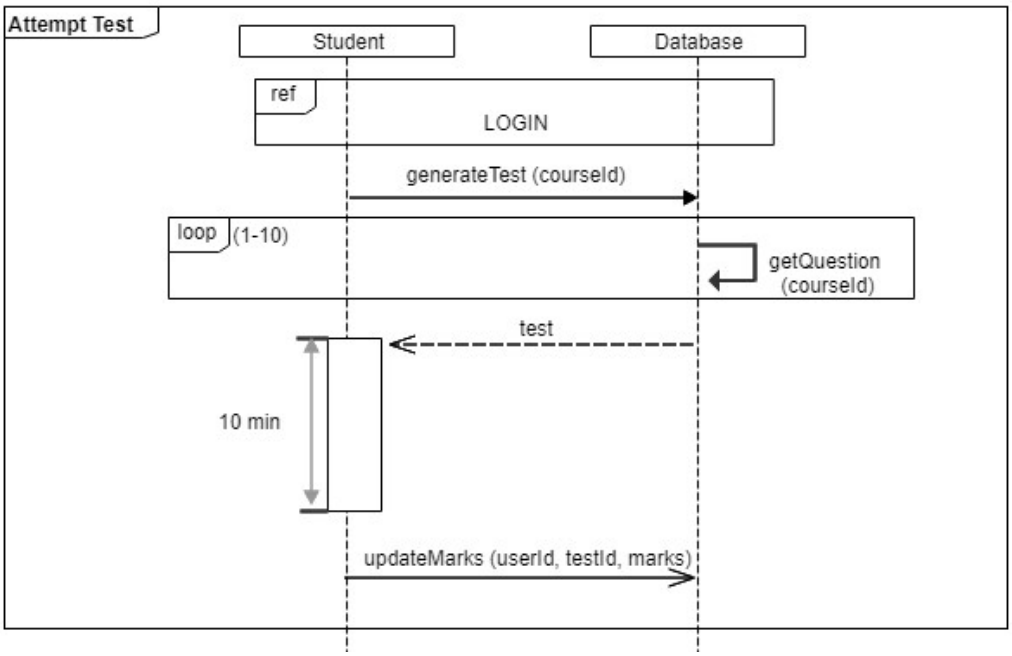


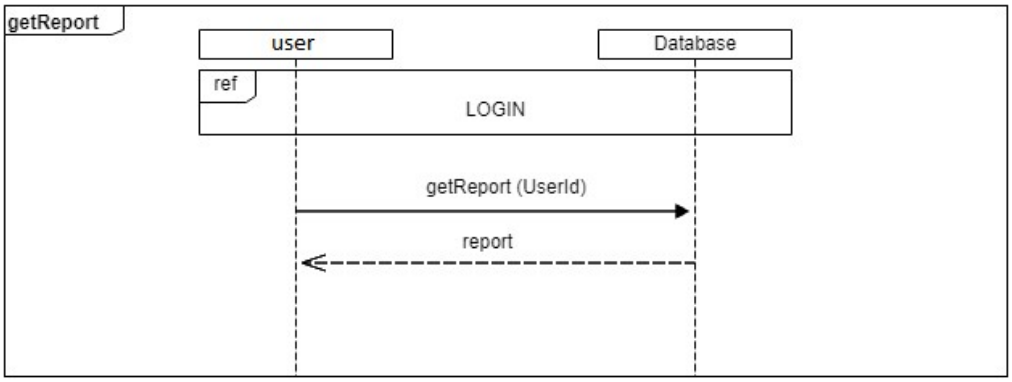
2.1.1.2 Class Diagram

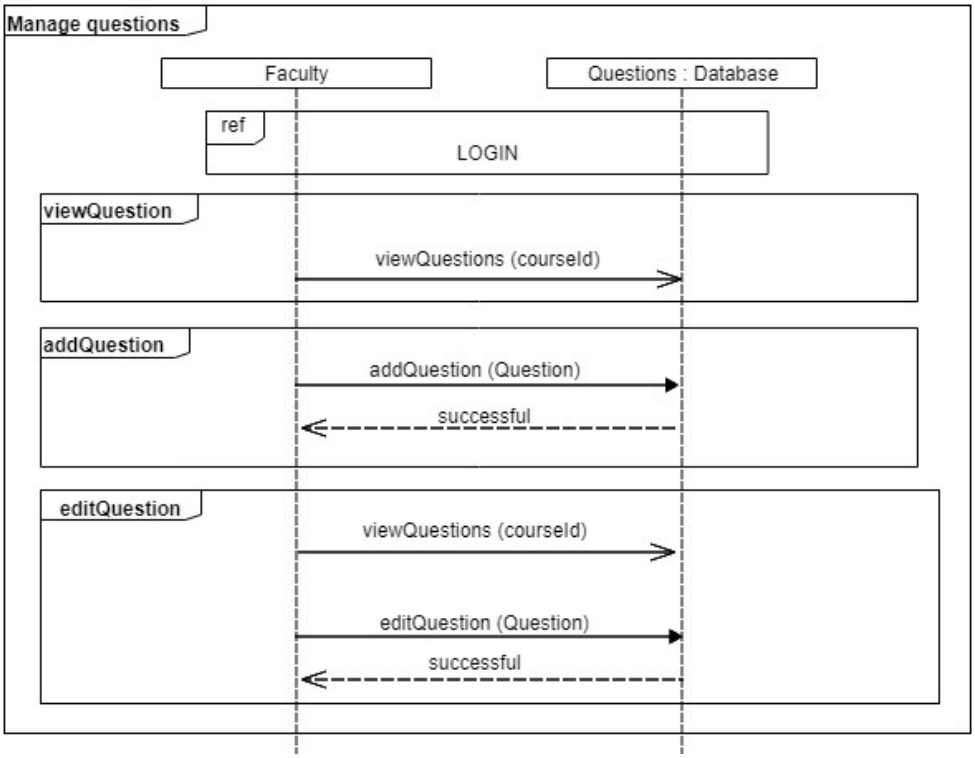


2.1.1.3 Sequence Diagram

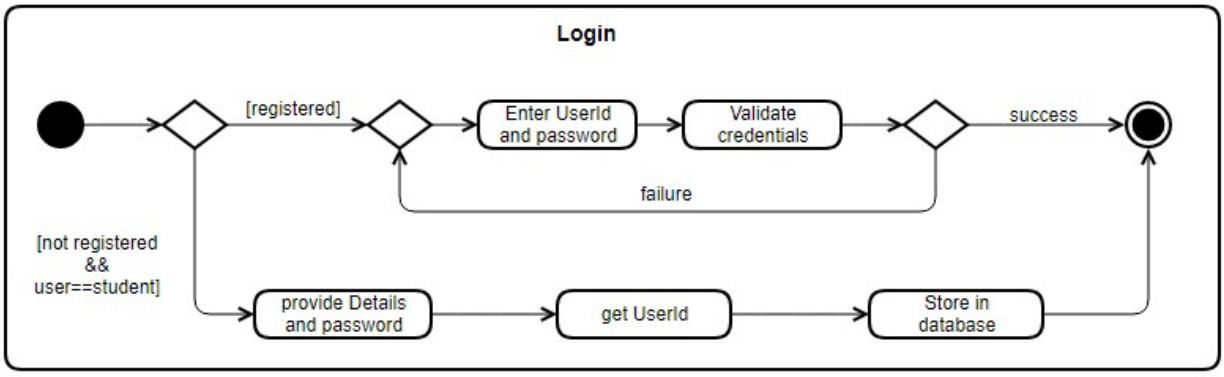


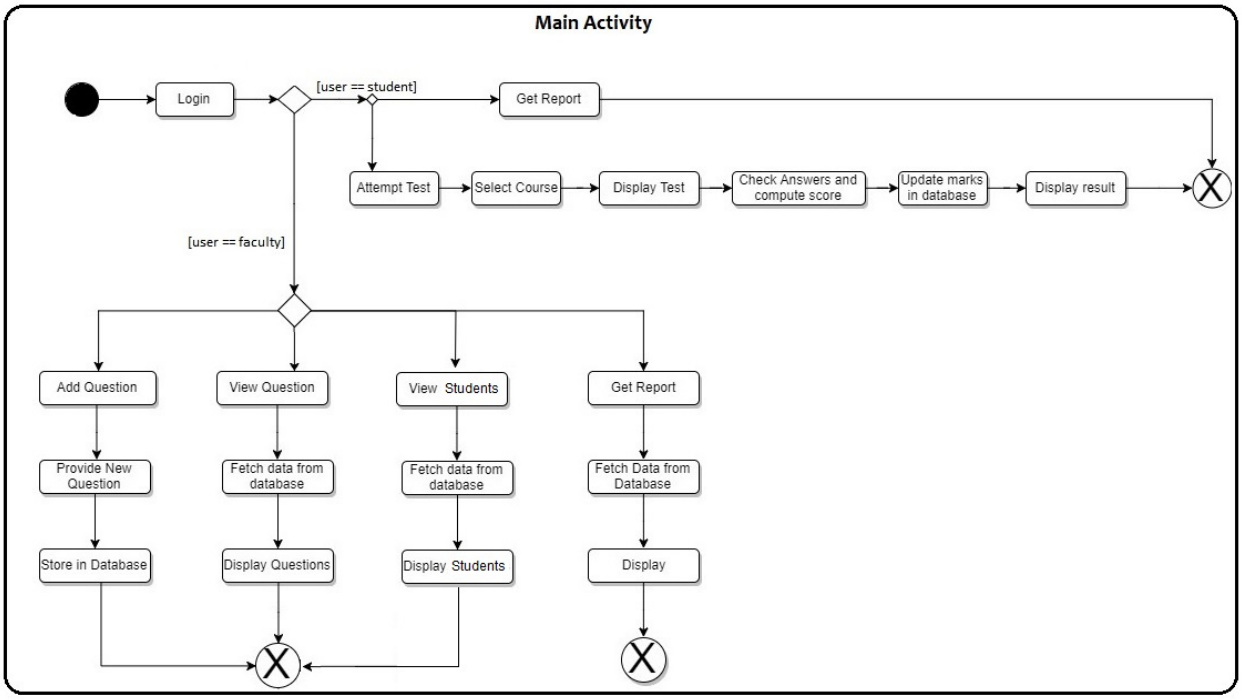

Attempt Test
Student
Database
ref
LOGIN
generateTest (courseId)
loop
(1-10)
getQuestion
(courseId)
test
10 min
updateMarks (userId, testId, marks)


getReport
user
Database
ref
LOGIN
getReport (UserId)
report


Manage questions
Faculty
Questions : Database
ref
LOGIN
viewQuestion
viewQuestions (courseId)
addQuestion
addQuestion (Question)
successful
editQuestion
viewQuestions (courseId)
editQuestion (Question)
successful

2.1.1.4 Activity Diagram





2.1.2 Non-Functional Requirements

- The computer system must be provided with the proper software/platform so as to run the software.

2.2 User Characteristics

There are two types of users – Student and Faculty who will interact with the software.

- All the users have unique login credentials.
- Student will select the course and attempt the MCQs test for that course. He will be then able to view the Score Sheet after the test is conducted
- Faculty will add, edit or delete the questions.
- All users have proper basic knowledge about operating the system and functionalities of the software.

3. Specific Requirements

3.1 Use case description

3.1.1

Name	Login
Short description	The user will provide valid username and password to login in the software
Actors	Professor, Student
Precondition	User has the username and password
Postcondition	The system accepts the username and password and user is logged in
Trigger	Launching of the software
System state in the event of an error	Display error message
Standard process	1. The user enters username and password 2. The credentials are verified from the database 3. If valid, user is logged in and if not valid, error message is shown
Alternative process	1. If the users have no username and password 2. User (only student) creates his own new account 3. Go to standard process

3.1.2

Name	Register
Short description	The user will be registered in the database
Actors	Student
Precondition	Users have all the required information
Postcondition	Users are registered
Trigger	Clicking on Register button
System state in the event of an error	Display error message
Standard process	1. The user enters all the required information 2. The credentials are stored in the database 3. User is registered
Alternative process	1. If the users did not enter all the required information 2. Prompt users to enter all the required information 3. Go to standard process

3.1.3

Name	Attempt test
Short description	The user (student) can attempt for the test
Actors	Student
Precondition	Users must be logged in
Postcondition	Students submit their answer
Trigger	Logging in the software and Clicking on attempt test
System state in the event of an error	Display error message
Standard process	1. The user logs in the software 2. The user wants to take the exam 3. User search the course of which he/she wants to take a test of. 4. User attempts the test 5. Finish in the given time 6. Submit the answer
Alternative process	1. If the user did not finish all the given questions in given time 2. User have to submit what they answered 3. Go to standard process

3.1.4

Name	Get Report
Short description	The test report is generated by calculating the score of submitted answers by the student
Actors	Professor, Student
Precondition	All authorised must be logged in
Postcondition	The test reports are displayed
Trigger	Logging in the software
System state in the event of an error	Display error message
Standard process	1. The user logs in the software 2. The user wants to get the test reports 3. The test reports are fetched from the database 4. All the tests taken by a particular user are displayed
Alternative process	1. If the student has not taken any test 2. Prompt the student to take a test

3.1.5

Name	Manage Questions
Short description	Displaying, adding, editing questions according to the courses
Actors	Professor
Precondition	All authorized must be logged in
Postcondition	The questions are displayed or added or edited
Trigger	Logging in the software by professors
System state in the event of an error	Display error message
Standard process	1. Professor wants to manage questions 2. Professor manages the question by their course ID. 3. The questions are managed
Alternative process	1. If professor fails to manage questions, he/she should contact database manager

3.2 Supportability

3.2.1 Naming Convention

All code will be written as specified by the Hungarian Naming Convention.